

RAGによる既知障害情報の活用方法に関する考察

Considerations of the Utilization of Known Failure Information Using RAG

研究員：手倉森 祥太 (株式会社日立システムズ)
横内 雅人 (株式会社日立ソリューションズ・クリエイト)
氏家 弘貴 (キヤノンメディカルシステムズ株式会社)
狩野 薫 (リコーITソリューションズ株式会社)
玉井 安明 (株式会社デンソー)
主査：石川 冬樹 (国立情報学研究所)
副主査：徳本 晋 (富士通株式会社)
アドバイザー：栗田 太郎 (ソニー株式会社)

研究概要

ソフトウェア開発の複雑化と短期化が進む中、障害発生リスクの増大が課題となっている。開発現場では既知障害情報を効率的に活用し、迅速な対策と再発防止を行うことが求められている。その手法として、大量のデータを学習し、人間の言語処理に近い応答を生成する生成AI (Generative AI) と、それに情報検索機能を付加する技術であるRAG (Retrieval-Augmented Generation) の活用が注目されている。本論文では、蓄積した既知障害情報のデータ形式や構造がRAGの検索性能および回答性能に与える影響について考察する。

1. はじめに

生成AIは、LLM (Large Language Model) を核として構成される。LLMは、膨大な量の学習データのトレーニングを通じて、パターン認識や文脈理解を行い、人間に近い言語処理を実現する能力を持つ。しかし、LLMの知識は学習データの範囲内に限定されており、未学習の情報や特定の専門分野に関する詳細な質問には適切に回答できないという課題がある。また、社内で発生する障害事例などの機密性の高い情報や最新情報については一般的な学習データに含まれないため、LLM単独では適切な回答を行うことが困難である。この課題を解決する技術としてRAGが注目されている^[1]。RAGは生成AIに情報検索機能を統合した技術であり、外部データベースや独自の情報リポジトリを利用して、検索結果に基づく適切な応答を生成する。この技術により、従来の検索システムと比較して高精度かつ迅速な問題解決が可能となる。

本研究では、障害対応へのRAG活用方法について実験と考察を行い、その有効性を検証した。

2. 背景

ソフトウェア開発における障害情報は、テキストや画像など様々な形式で存在する。従来のキーワードベースの検索システムでは、情報の文脈や関連性を十分に考慮できず、検索結果が不完全または無関係な情報を含む可能性があり、必要な情報を迅速に抽出するには限界がある。この問題を克服する技術としてRAGが注目されている。RAGは、生成AIが外部データソースから関連情報を検索し、その情報を基に応答を生成する技術である。検索エンジンと生成AIの組み合わせにより、障害情報について迅速かつ正確な解決策を提示することが期待される。RAGの精度向上には、検索性能と回答性能の両方が重要であり、いずれも既知障害情報の品質に影響を受ける可能性がある。一方で、社内には多数の障害情報が蓄積されており、これらを整備することの効果についての検証が求められる。さらに、生成AIを最大限に活用するためには、適切なプロンプト設計が求められ^[2]、この点もRAGの活用における課題である。

障害対策に生成AIを活用する際の課題は以下の通りである。

まず、生成AIの回答性能において、障害情報が十分に与えられていない場合、その出力結果の信頼性に影響を与える可能性がある (RQ1)。次に、RAGの検索性能が低いと関連性の低い情報が返され、それに基づく不適切な出力が生成されるリスクがある (RQ2)。さらに、障害情報は複数の現象を含む場合に、これらの情報を適切に処理して出力結果に反映させる方法についても検証が必要である (RQ3)。最後に、RAGにおけるデータの形式や構造が、検索や回答生成の精度に与える影響を定量的に評価する必要がある (RQ4)。

こうした背景を踏まえ、本研究における研究課題を以下の通り定める。

RQ1：LLMに障害情報を含めることで、出力結果に対しどのように影響するか

RQ2：RAGにおける検索性能によって、出力結果に対しどのように影響するか

RQ3：障害情報内に複数の現象が含まれる場合、出力結果に対しどのように影響するか

RQ4：RAGに蓄積したデータの形式と構造によって、出力結果に対しどのように影響するか

3. 実験

3.1 実験方法

本研究では、上記研究課題に基づき、障害発生事例に対し生成AI（LLM及びRAG）の有効性を実証するための実験を実施する。実験は以下の手順で行う。

3.1.1 実験データ

IPAが公開する情報処理システムにおける障害情報ファイル53件^[3]を用いる。このファイルには、各情報処理システム障害の再発防止に至るまでの情報（発生現象、原因、対策、対策の効果、教訓）がまとめられており、そのまとめ方はテキスト文章、画像、グラフ化等で表現されている。ファイル形式はPDF形式とテキスト形式で用意した。この障害情報ファイルを利用して各RQの実験を進める。

3.1.2 実験環境の設定

実験環境は、検索システム：Azure AI Search、検索方式：セマンティック検索、検索スコアの計算アルゴリズム：BM25、LLM：ChatGPT-4oで設定する。

3.1.3 実験の設定

各実験の設定は次の通りである。

(1) 質問文：障害発生時の典型的な質問を設定し質問内容は固定。各RQのプロンプトを参照。

(2) 実験環境パラメータの固定：実験環境は全RQで一貫して固定。外部要因を最小限とする。

3.2 実験の実施

実験は4つの研究課題に基づいて実施される。各実験内容は以下の通りである。

【実験1 (RQ1)】

LLMを活用すれば、障害発生時の情報から原因と対策を的確に特定・提示できると考える。本仮説を検証するため、LLMに①障害情報を与えない場合と、②障害情報を与えた場合の2条件で、原因と対策の提示が可能かを検証する。

(1) 実験方法

以下形式のプロンプトをLLMに与えて、障害情報の原因と対策方法を取得する。

①障害情報を与えない場合

(障害の現象)の原因を教えてください。その原因に対する対策を教えてください。

②障害情報を与えた場合

以下の事例をもとに、(障害の現象)の原因を教えてください。そして、その原因に対する対策を教えてください。

■事例：(障害の問題, 障害の原因, 障害の対策等)

※カッコ内は具体的な事象や状況に応じて変化する

(2) 評価基準

LLMの回答を基に、障害情報の原因及び対策の取得可否を評価する。

[基準1]：質問されたことに全て回答し、原因と対策が明確か

[基準2]：原因と対策が論理的に対応しているか（因果関係の明確さ）

(3) 実験結果

全実験データ53件に対して基準1, 2の観点で評価した。

[基準 1]：質問されたことに全て回答し，原因と対策が明確か

①障害情報を与えない場合：回答には原因と対策が記載され，質問に求められた内容を全て満たしていた。ただし，回答内容は一般的な知識から学習した内容となっていた。

②障害情報を与えた場合：回答には原因と対策が明確に記載され，求められた内容に全て回答していた。プロンプトで与えた障害情報に基づく具体的な対策も含まれ，具体的な情報が提示されていた。回答にはプロンプトで与えた障害情報に基づく原因と対策が明確に記載され，求められた内容に全て回答していた。

[基準 2]：原因と対策が論理的に対応しているか（因果関係の明確さ）

①障害情報を与えない場合：回答された原因と対策には論理的な結びつきがなく，内容が抽象的で因果関係が明確でない傾向がみられた。

②障害情報を与えた場合：回答された原因と対策には論理的結びつきがあり，内容が具体的に因果関係が明確であった。

(4) RQ の検証

①障害情報を与えない場合と②障害情報を与えた場合のいずれも，原因および対策の提示は可能であった。特に②では，障害情報の提示によって具体的な回答が得られた。障害対策における LLM の有効性を高めるためには，LLM により具体的な情報を与えて回答させることが重要と考えられる。

【実験 2 (RQ2)】

RAG はプロンプトから検索 Query を生成し検索を行うが，データ形式が異なる場合は RAG の検索 Query に差が生じることは既知の事実^[5]である。そこでプロンプトとデータ形式によって検索 Query の構成を分析することで，検索結果に与える影響を確認する。

(1) 実験方法

実験では，障害情報の事例 G11 「システムの運用・保守に関する教訓」を使用し，以下形式のプロンプトを用いる。

保守作業での設定ミスで待機系側に必要ファイルが存在せず，障害時に切替失敗の原因を教えてください。その原因に対する対策を教えてください。

(2) 評価基準

データ形式による検索 Query の差異を確認し，その差が回答に与える影響を確認する。

(3) 実験結果

プロンプトに入力した内容から生成された検索 Query に使用されるキーワードは，図 3 のように 3 つのカテゴリ「一般的なキーワード」「特徴的なキーワード」「文章的なキーワード」に分類される。事例 G11 のプロンプトにおいて，「原因」に対する回答文にはプロンプトのキーワードが使われやすく，「対策」の回答文には使われにくい傾向が見られた。表 1 はプロンプトに入力した文章が検索 Query に分解された結果から，その検索 Query に対する考察を整理した。

表 1. プロンプトに対する検索 Query の生成結果とその結果から得られる考察

プロンプトへ入力した事例	検索 Query	検索 Query 発生回数		考察	
		テキスト	PDF	検索 Query	事例との一致性
保守作業での設定ミスで待機系側に必要ファイルが存在せず障害時に切替失敗した事例の原因と対策を教えてください	保守作業	1/5回	4/5回	特徴的なキーワード	具体的事例と一致し易い
	設定ミス	1/5回	4/5回		
	必要ファイル	1/5回	4/5回		
	不存在	1/5回	4/5回		
	障害時	1/5回	4/5回	一般的なキーワード	一般的事例と一致し易い
	切替失敗	1/5回	4/5回		
	原因	1/5回	4/5回		
	対策	1/5回	4/5回		
	待機系側	1/5回	3/5回	特徴的なキーワード	具体的事例と一致し易い
	事例	1/5回	4/5回	一般的なキーワード	一般的事例と一致し易い
	待機軽ファイル不足	1/5回	2/5回	特徴的なキーワード	具体的事例と一致し易い
	設定ミスによる切替失敗	4/5回	1/5回	文章的なキーワード化	「原因」とは一致し易い 「対策」とは一致し難い
	待機系に必要なファイルがない	4/5回	1/5回		
	保守作業での切替失敗	4/5回	1/5回		

テキスト形式では検索 Query は主に「文章的なキーワード」で構成され、PDF 形式の検索 Query では「一般的なキーワード」が多く使用されることが分かった。事例 G11 は、プロンプトに特徴的なキーワードが少ない事例であり、テキスト形式と PDF 形式では結果が異なった。

(4) RQ の検証

データ形式の違いにより同じプロンプトから生成される検索 Query に違いがあることが確認された。この差は、検索 Query 生成時に LLM がプロンプトから検索 Query を最適化する過程で、DB 情報を参照することに起因している。テキスト形式ではキーワードが文章的であり、検索結果と事例内容との一致性は高くなる傾向がみられた。PDF 形式ではキーワードが単語的になり、G11 以外の事例に含まれるキーワードで検索されるため、事例の原因と対策が回答に反映されにくくなる。この結果から、プロンプトに特徴的なキーワードが少ない場合、ドキュメント内に含まれる特徴的な要素(専門的な用語や文章)との一致度が低くなることが分かった。また、生成される Query の長さも検索性能に影響を与えることが示された。検索対象となるドキュメントの種別や構造によって、検索 Query の質が変わるため、検索性能を高めるには適切なデータ形式を選択することが重要である。

【実験 3 (RQ3)】

(1) 実験方法

障害情報内に複数現象が記載されている事例 G11 を利用し、各現象に関連する質問を行い、その出力結果から、原因と対策内容がデータ形式や検索 Query の相違点を確認する。

(2) 評価基準

LLM の回答を基に、障害情報の原因及び対策の取得可否を評価する。

[基準]：現象①～③を含む質問に対して、返された回答の正確さをスコアで評価する。

(2点：G11 の障害情報のみ検出，1点：G11 を含む複数の障害情報を検出，0点：G11 未検出)

(3) 実験結果

以下は、現象①～③を含むプロンプトを使った試行を、各 5 回実施したスコアである。現象②のスコアが最も高く、回答性能が高く優れたプロンプトであることが分かった。

表 2. 現象①～③別のスコア

現象	プロンプト内容	実行回数	PDF		TXT		
			原因	対策	原因	対策	
			評価	合計	評価	合計	
①	保守作業での設定ミスで待機系側に必要ファイルが存在せず、障害時に切替失敗	1回目	×	×	○	×	0
		2回目	×	×	×	×	
		3回目	○	×	×	×	
		4回目	×	×	○	×	
		5回目	×	×	×	×	
②	ディスクモジュールの自己診断機能で、異常検知のみで機能停止する仕様となっていた事例の、原因と対策を教えてください	1回目	○	○	○	○	10
		2回目	○	○	○	○	
		3回目	○	○	○	○	
		4回目	○	○	○	○	
		5回目	○	○	○	○	
③	保守作業チェックや修正プログラムの適用がシステムによらず一律となっており、重要なミスを見落とした事例の、原因と対策を教えてください	1回目	○	○	○	○	6
		2回目	△	△	△	×	
		3回目	○	○	△	△	
		4回目	△	○	○	△	
		5回目	△	△	○	○	

○：G11の障害データのみ検出(2点)

△：G11を含む複数の障害データを検出(1点)

×：G11未検出(0点)

各プロンプトの検索性能の差異を確認する。以下は、検索 Query で使われたキーワード別の出現結果である。キーワードの並びは検索結果の影響を表し、左側ほど影響が高い。また、○は出現、－は出現なしを意味する。

表 3. 現象①の検索 Query で使われたキーワード別出現状況

プロンプト内容	データ形式	試行回数	出現回数															
			作 業	保 守	ミ ス	設 定	必 要 フ ァ イ ル	不 存 在	障 害 時	失 敗	切 替	原 因	対 策	系 側	待 機	事 例	フ ァ イ ル 不 足	待 機 系
保守作業での設定ミスで待機系側に必要ファイルが存在せず、障害時に切替失敗した事例の原因と対策を教えてください。	PDF	5回	4	4	4	4	4	4	4	4	4	4	3	4	2	1	1	1
	TXT		1	1	1	1	1	1	1	1	1	1	1	1	0	4	4	4

- ※1 設定ミスによる切替失敗の原因と対策
- ※2 待機系に必要なファイルがない場合の切替失敗事例
- ※3 保守作業での切替失敗の原因と対策

表 4. 現象②の検索 Query で使われたキーワード別出現状況

プロンプト内容	データ形式	試行回数	出現回数								
			モ ジ ュ ー ル	デ ィ ス ク	機 能	自 己 診 断	異 常 検 知	機 能 停 止	事 例	原 因	対 策
ディスクモジュールの自己診断機能で、異常検知のみで機能停止する仕様となっていた事例の、原因と対策を教えてください	PDF	5回	5	5	5	5	5	5	5	5	0
	TXT		5	5	5	5	5	5	5	1	

表 5. 現象③の検索 Query で使われたキーワード別出現回数

プロンプト内容	データ形式	試行回数	出現回数												
			対 策	原 因 と ミ ス	重 要 な	見 落 と し	事 例	チ ェ ッ ク	保 守 作 業	グ ラ ム	修 正 プ ロ グ ラ ム	適 用	一 律	シ ス テ ム 保 守	※1
保守作業チェックや修正プログラムの適用がシステムによらず一律となっており、重要なミスを見落とした事例の、原因と対策を教えてください	PDF	5回	4	4	1	4	4	4	3	4	3	1	1	1	0
	TXT		3	2	0	3	3	3	2	0	0	3	2	0	3

- ※1 保守作業チェックのミス事例と原因
- ※2 修正プログラムの適用ミスの原因と対策
- ※3 システムメンテナンスにおける重要ミスの防止策
- ※4 システム保守作業の見落とし原因と対策

現象②のプロンプトには、他のプロンプトに比べ検索に使われたキーワード数が少ない。更に特徴的なキーワード「ディスクモジュール」や「自己診断機能」が、現象②の検索結果に影響を与えることが分かった。

(4) RQ の検証

質問に使用する現象によって、回答結果に差異があった。現象②の概要には他の障害情報に含まれない特徴的なキーワード「ディスクモジュール」や「自己診断機能」が含まれ、回答精度に影響を与えていた。プロンプトを設計する際に、特徴的なキーワードを用いることが回答精度を上げる上で重要と考える。

【実験 4 (RQ4)】

(1) 実験方法

RAG に蓄積したデータ形式と構造の違いが出力結果の精度に影響すると仮説を立てて検証する。PDF(非構造化データ), L1(PDF をコピー&ペーストしたテキストデータ), L2(L1 を整形したテキストデータ), L3(L2 をタグ付けした構造化データ)を使用し, その4つのデータに対して RAG 出力結果を多重比較する。

出力結果の精度を幅広く検証するため, 検証に使用するプロンプトは RAG の評価フレームワーク RAGAS (Retrieval Augmented Generation Assessment) v2.0^[4]のテストセット生成機能を利用して生成する。テストセットには障害情報に関連する観点(原因/対策/効果など)を持つペルソナを設定し, 質問には, 5つのカテゴリ(問題, 原因, 対策, 教訓, 再発防止策)に基づくランダムな質問 83 件を使用する。

(2) 評価基準

評価には RAGAS が提供するメトリクス (Context Precision, Context Recall, Context Entities Recall, Faithfulness, Answer Relevance)を用いる。検索性能の指標として, Context Precision は Context 内の関連チャンクの割合がどの程度か, Context Recall は関連ドキュメントがどれだけ取得されたか, Context Entity Recall は正解に Context の Entity がどれだけ含まれているかを表す。回答性能の指標として, Faithfulness は生成された回答が RAG によって与えられた Context に基づいているか, Answer Relevance は生成された回答が元の質問にどれだけ適切であるかを表す。それぞれ 0~1 の小数で評価され, 値が高いほど性能が高いとされる。

同一の質問に対して, PDF, L1, L2, L3 を蓄積した環境でレスポンスを取得し, 各メトリクスの値を比較した。

(3) 実験結果

RAGAS を用いた実験結果を表 1 に示す。データ形式や構造が検索性能や回答性能に与える影響を比較した結果, データ形式や構造は, 検索性能 (Context Precision, Context Entity Recall)に影響を与えるが, 回答性能には大きな影響を及ぼさないことが確認された。

Context Precision に注目すると, PDF が L2/L3 より有意に高い値を示した ($p < 0.05$)。これは PDF を検索に使用することで回答生成に寄与することを示している。一方, Context Entity Recall では, L1 が PDF/L2/L3 より有意に低下した ($p < 0.05$)。文の区切りを無視したドキュメントを検索対象とすると, 回答に必要な単語やキーワードを網羅しにくいことがわかる。

一方で Answer Relevancy と Faithfulness については各群で有意差が見られず, 検索対象の違いが回答生成の性能に影響を与えないことが確認された。Faithfulness は全て 0.8 以上で, 取得した Context に基づく回答性能が高いと言える。Context Recall も同様で, 取得した Context が正解データに近いことを示している。

表 6. データ形式/構造による RAGAS 指標の一元配置分散分析結果: Mean (SD)

データ形式	id	Context Precision (N = 83)	Context Entity Recall (N = 83)	Context Recall (N = 83)	Answer Relevancy (N = 83)	Faithfulness (N = 83)
data1	PDF	0.715 (0.104)	0.347 (0.055)	0.803 (0.117)	0.678 (0.020)	0.887 (0.036)
data2	L1	0.635 (0.110)	0.245 (0.050)	0.840 (0.103)	0.673 (0.022)	0.886 (0.037)
data3	L2	0.580 (0.116)	0.341 (0.052)	0.826 (0.108)	0.685 (0.019)	0.865 (0.038)
data4	L3	0.582 (0.127)	0.335 (0.059)	0.833 (0.117)	0.685 (0.017)	0.883 (0.034)

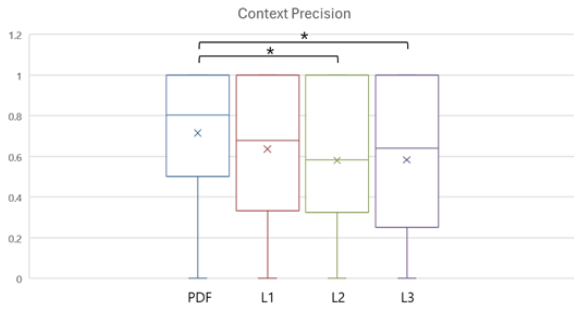


図1. Context Precision の分布図

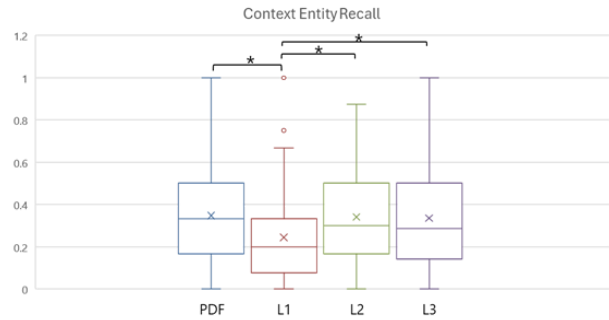


図2. Context Entity Recall の分布図

(4) RQ の検証

RAGAS Score では、Context Precision において PDF が L2/L3 より有意に高い値を示し、検索性能の評価として PDF が優れている結果となった。これは、PDF が L2/L3 に比べて検索 Query が単語となりやすく、一致度の高いドキュメントを検索し易いためと考えられる。Context Entity Recall は、PDF/L2/L3 が L1 より有意に高く、L1 のように文や単語区切りを考慮しない構造では、目的語を RAG が正確に認識しにくいことが判明した。これは、L1 の Context が文節や単語区切りではないことで、PDF/L2/L3 と比べて評価指標が低下していることが原因と考えられる。また、Context Entity Recall の値が全体的に 0.2~0.3 と低いのは、テキスト化時に含まれる単語間のスペースや、LLM 側の Entity 作成単位に依存していること、さらに User Input の内容によって参照する Context が不適切と判定されるケースを含むことが影響している。

検索性能を向上するには、適切な単語や文節単位で区切られているドキュメントの使用や、ドキュメント自体の文章を適切に区切ることのできる処理をシステムに組み込むことが有効と考えられる。検索性能では PDF が優れる結果となったが、Answer Relevancy や Faithfulness の有意差が見られなかったことから、検索性能の低下が本実験環境での回答性能に与える影響は小さいことが示された。影響が少なかった要因として、セマンティック検索による Top 5 の Context の中に必要なドキュメントが大部分含まれていたこと、更に回答生成を行う LLM が高性能であり、限られた Context でも正確な推論が可能だったことが挙げられる。

4. 考察

一般的に障害情報は問題(内容)、原因、対策で構成されており、その構成を持つ具体的な障害情報を利用することで、LLM を活用し、障害対応時に「原因」と「対策」をより具体的に取得できる可能性があることが RQ1 の実験結果により示唆された。RQ2 の結果では、数値的な性能評価により、PDF 形式の方が検索性能は良く、単語/文章の区切りを考慮しないドキュメントではいずれのデータ形式/構造よりも検索性能が下がることが分かった。しかし、回答性能への影響を少なくする要素を組み込むことで、一定の回答性能は維持されることが確認できた。

RQ3, 4 の実験結果から、プロンプトには既知情報に特徴的なキーワードを含めた方が検索精度は高く、文章的にする場合はキーワードを含んだ文章の区切り入れると、一致度の高い文章を得やすいことが確認された。一部のテスト事例では、プロンプトに特徴的なキーワードを含まないことで生成される検索 Query が特定ドキュメントを探し難くなることが確認された。このため、DB 上でドキュメント量が増加すると、検索 Query 生成による意図するドキュメント取得が難しくなることが推察された。この対策としては、ドキュメント作成時に障害情報に関する特徴的なキーワードを含むように記載すること、また Graph-RAG として言葉の関連性を高めること、他の検索要素(システム, Ver., 日付, 人等)を含めることで、検索対象の範囲を絞り込み、精度良く既知情報を得ることが可能になると考えられる。

本研究では検索対象とする既知の障害情報と障害事象を表す文章がセットで存在している前提で実験したが、現実の障害発生時には障害事象を表す正確な言葉や解決方法が存在しないことが想定される。この問題はRAGにおいても、従来のLLM使用におけるプロンプトエンジニアリングの問題が存在することを意味している。しかし研究結果より、ユーザープロンプトに障害情報の記述に特徴的な要素が少なくても原因部分に類似する内容を持つ既知データをRAGより取得し易いことが判明しているため、より特徴的なプロンプトを生成する仕組み(取得ドキュメントまたはDB上ドキュメントからより特徴的なキーワードを取捨選択してプロンプトを再作成するなど)を導入することで、ユーザーが正確なプロンプトを記述しなければならないという問題を低減することができると考えられる。このことから、抽象的なプロンプトを入力してもユーザーが必要とする障害対応に役立つドキュメントや情報を、精度よく入手できると考えられ、RAGを用いた既知障害情報の取得は幅広いユーザーが利用可能であり、効果的な障害対応や予防を行うことで、製品品質が向上することが示唆される。

5. おわりに

本論文では、RAGに蓄積したデータ形式や構造が検索性能に与える影響について実験結果を示した。実験結果から、LLMを用いて適切なプロンプトを設定することで、類似する既知情報を入手することが可能であることが確認できた。また、データ形式や構造に違いがあると検索性能は低下するが、回答性能には影響しないことが分かった。

これにより、本研究はRAGを使用して既知の障害情報から関連情報を迅速に抽出し、障害対策に活用することが、開発現場の品質向上及び効率化が有効となることを示した。今後はデータ量による影響や精度向上の方法を研究し、更なる進展を図る。

謝辞

本論文の執筆にあたり、石川冬樹主査、徳本晋副主査、栗田太郎アドバイザーには丁寧なご指導を賜りました。深く御礼を申し上げます。

参考文献

- [1] Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,
<https://arxiv.org/abs/2005.11401> (2025/01/22 参照).
- [2] 生成 AI によるソフトウェアパターン適用の試行 ～ソフトウェア設計品質向上に向けたプロンプトの考察～,
https://www.juse.or.jp/sqip/workshop/report/attachs/2023/5_AI_Design_Quality_%E3%83%81%E3%83%BC%E3%83%A0_%E6%9C%AC%E6%96%87.pdf?20240319.
- [3] 情報処理システム高信頼化教訓のリンク集 (IT サービス編) | アーカイブ | IPA 独立行政法人 情報処理推進機構,
<https://www.ipa.go.jp/archive/digital/iot-en-ci/system/lesson.html> (2025/01/22 参照).
- [4] RAGAS,
<https://docs.ragas.io/en/stable/> (2025/01/22 参照).
- [5] Revolutionize your Enterprise Data with ChatGPT: Next-gen Apps w/ Azure OpenAI and Cognitive Search | Microsoft Community Hub,
<https://techcommunity.microsoft.com/blog/azure-ai-services-blog/revolutionize-your-enterprise-data-with-chatgpt-next-gen-apps-w-azure-openai-and/3762087>
(2025/01/27 参照).